First Hit

☑ Generate Collection Print

L3: Entry 2 of 29 File: PGPB Oct 17, 2002

DOCUMENT-IDENTIFIER: US 20020152328 A1
TITLE: Network adapter with shared database for message context information

Summary of Invention Paragraph:
[0016] Although the preferred embodiments described herein relate specifically to IB switch fabrics and to HCAs used with such fabrics, the principles of the present invention may similarly be applied, mutatis mutandis, to channel adapters of other types, such as target channel adapters (TCAs), as well as to network interface adapters used in other packet networks. Similarly, the notion of sharing a local database among multiple service instances can be used not only for recording outstanding messages, but also for managing other transport context records of variable size. For example, such a local database may be used to hold scatter lists, indicating the locations in a host memory to which data from incoming read response and send request messages should be written.

Detail Description Paragraph:
[0055] In order to send out packets from HCA 22 on a given QP over network 26, host 24 posts WQEs for the QP by writing work request descriptors 44 in memory 38 (FIG. 1). For RDMA write and send requests, the descriptors indicate the source of data to be sent and its destination. The data source information typically includes a "gather list," pointing to the locations in memory 38 from which the data in the outgoing message are to be taken. For RDMA read requests, the descriptors include a "scatter list," indicating the locations in memory 38 to which the requested data are to be written, when the read response message is received. After host 24 has prepared one or more descriptors, it "rings" a doorbell 50 of HCA 22, by writing to a corresponding doorbell address occupied by the HCA in the address space on the host bus. The doorbell thus serves as an interface between host 24 and HCA 22. The process of ringing and servicing doorbell 50 is described in detail in the above-mentioned patent applications.

Detail Description Paragraph:
[0056] In response to the doorbell being rung, an execution unit 52 queues the QPs having outstanding WQEs, and schedules the QPs for service by arbitrating among the queues. When a QP reaches the head of its scheduling queue, the execution unit parses each of the WQEs in the linked list belonging to that QP. Based on the WQEs, the execution unit prepares gather entries to submit to a send data engine (SDE) 54. Each gather entry defines a piece of data to be incorporated in a packet to be prepared and sent out by the SDE. SDE 54 gathers the data to be sent from the locations in memory 38 specified by the WQEs and places the data in output packets for transmission over network 26. The data packets prepared by SDE 54 are passed to an output port 56, which performs data link operations and other necessary functions, as are known in the art, and sends the packets out over network 26.

Detail Description Paragraph:
[0057] Packets sent to HCA 22 over network 26 are received at an input port 58, which similarly performs data link and buffering functions, as are known in the art. A transport check unit (TCU) 60 verifies and processes transport-layer information contained in the incoming packets, as described in greater detail hereinbelow. When the TCU receives a packet containing data to be written to memory 38, it passes the data to a receive data engine (RDE) 62. The RDE serves as a

scatter engine both for RDMA write and send requests received by HCA 22 from remote requesters (such as HCA 28, in FIG. 1) and for RDMA read responses returned to HCA 22 by remote responders (HCA 28 or TCA 32, for example). A translation and protection table (TPT) 64 acts as a host interface in HCA 22. It is used for address translation and protection checks to control access to memory 38 by elements of HCA 22, including execution unit 52, SDE 54 and RDE 62, as well as by other, remote entities connected to network 26.

Detail Description Paragraph:
[0065] Scatter list pointer--gives the location of the scatter list for RDMA read requests. The scatter list itself indicates the address range in memory 38 to which RDE 62 should write the data contained in the read response packets. Preferably, the scatter list is held in a separate memory array (not shown) that is allocated to the RDE for this purpose. This array is also preferably shared among the QPs that have read request messages outstanding, in a manner similar to the sharing of LDB 68.

Detail Description Paragraph:
[0075] Returning now to FIG. 3, a linklist controller 74 is responsible for maintaining the pointer structure of memory 72. The pointers are used by a read/write engine (RWE) 76 in pushing entries onto the tails of the lists and popping entries from the heads, in response to requests from execution unit 52 and TCU 60, respectively. When the execution unit provides a new message record to be added to the linked list of a given QP in memory 72, controller 74 checks free head pointer 100 to find the free entry 92 to which this new record is to be written. Preferably, LDB 68 issues credits to the execution unit to indicate that there are free entries available, and the execution engine will not process a WQE to generate a request message unless it has such a credit. Pointer 94 of free entry 92 points to the next free entry in the free list. Controller 74 updates free head pointer 100 so that it points to this next free entry, which now becomes the head of the free list. Meanwhile, RWE 76 fetches tail pointer 98 for the given QP from send context 67, via a QP context (QPC) interface buffer 82. Controller 74 uses pointer 98 to find the last record in the linked list for this QP, and updates pointer 94 of this last record (which previously had a null value) so that it now points to the new record that has just been written. It also instructs RWE 76 to update the value of tail pointer 98 in the send context 67 so that it points to the new record that has just been added to the end of the list.

Detail Description Paragraph:
[0078] Although the operation of LDB 68 is described above with reference particularly to its function in maintaining records of outstanding messages, the LDB can be viewed more generally as a cache of flexible size, which operates in conjunction with cache memory 66. While each QP receives an allocation of fixed size in memory 66, the QP allocations in LDB 68 grow and shrink in response to demand for each QP. The LDB can be shared among multiple QPs not only for recording outstanding messages, but also for managing other transport context records of variable size, such as scatter lists for use in handling data from incoming read response and send request messages. Furthermore, although the description of the LDB given here uses specific vocabulary and conventions of IB fabrics and channel adapters, the principles of the present invention may similarly be applied to other types of networks and adapters that must serve multiple transport instances concurrently.